Fundamentals

Author	Dawson Lee
Last Update	2022/03/07

Introduction

Evolution of Processors

- 8086: x86 Architecture
- Pentium
- IntelCore

x86 Manufacturers

- Intel
- AMD
- VIA
- Others are acquired or discontinue

ISA

- x86 is CISC (complex instruction set computer)
    - x86 has varying length instructions
- RISC (reduced instruction set computer)

Die

Wafer consists of many Dies.

First Microprocessor

- Intel 4004
- Max CPU clock rate: 108 kHz

General

- Transistors doubles every 1.5-2.0 yrs
- Process speed doubles every 1.5-2.0 yrs
- DRAM size doubles every 1.5-2.0 yrs

Single Processor Performance

- 2003: from uni-processor to multi-processor
- Pipeline (ILP) reaches its limit
- Power limit
- Move to multi-processor
- Clock rate 1% growth

Effects

- enhanced capability
- lead to new class of computers
- dominance of microprocessor-based computers
- software development shift focus on productivity

Current Trends

- DLP
- TLP
- RLP

Classes of Computers

▼ class

1. Personal Mobile Device
2. Desktop Computing
3. Servers
4. Clusters / Warehouse Scale Computers
5. Supercomputers (require faster network than WSCs)
6. Embedded Computers

(refer to textbooks for details)

Flynn’s Taxonomy

- Single instruction stream, single data stream (SISD)
- SIMD (GPU, AVX extensions, vector arch)
- MISD (no commercial implementation)
- MIMD (tightly-coupled / loosely-coupled)

Two Kinds of Parallelism in Applications

- Data-Level Parallelism (operate on many data at the same time)
- Task-Level Parallelism (tasks can operate independently)
- example: web crawler
    - crawl web pages
    - parse HTML data (data-level)
    - run parse HTML task (task-level)

Four MajorWays for Exploiting Parallelism

- ILP (pipelining, data-level)
- Vector arch / GPU (data-level)
- Thread-Level Parallelism (multi-core, data-level or task-level)
- Request-Level Parallelism (clusters, data-level or task-level)

Define Computer Architecture

- old view: ISA
- ISA is same as programming model
- abstracts hardware and software interface

Contents of ISA

- register (size, number)
- class of ISA (register-memory, complex or load/store)
- memory addressing (byte addressing, little/big-endian)
- addressing modes
- instruction operands (e.g. RISC-V 3 operands, x86 only 2)
- available operations (e.g. RV32I doesn't support hardware mul and div)
- control flow instructions (e.g. ret or jalr in RISC-V reads return address from ra, but x86reads from stack and pop)
- instruction encoding (fixed length / variable length)
- etc.

Addressing Mode

- register, r4 $\leftarrow$ r4 + r3
- immediate, r4 $\leftarrow$ r4 + 3
- displacement, r4 $\leftarrow$ r4 + M[100+r1]
- register deferred, r4 $\leftarrow$ r4 + M[r1]
- etc. refer to textbook

MIPS ISA

- `| op 5 | rs 5 | rt 5 | rd 5 | shamt 5 | funct 6 |` (32 bits)

> op: operation
> rs: register first source
> rt: register second source
> rd: register destination
> shamt: shift amount
> funct: function code

Addressing Mode Examples

register addressing (no shamt)
immediate addressing (rd-funct as operand)
base (displacement) addressing (rd-funct as offset + base register)
pc-relative addressing (rd-funct as offset + PC)

Micro-architecture

- micro-arch, also called computer organization, is the way a given ISA is implemented on a processor.
- a given ISA can be implemented with different micro-archs

Concepts

- pipelining
- hierarchical memory organization
- cache
- cache coherence
- branch prediction
- super-scalar
- out-of-order execution
- register renaming
- multi-processing and multi-threading

Computer Architecture in General

- working in constraints
- market target?
- cost/performance?
- tradeoff in material and process
- Computer Architecture is about designing the organization and hardware to meet goals
- and functional requirements

Trends in Technology

Five Critical Implementation Techs

1. IC technology
2. semiconductor DRAM
3. semiconductor flash
4. magnetic disk technology
5. network technology

IC technology

vacuum tube $\rightarrow$ transistor $\rightarrow$ semiconductor (gates, memory cells, interconnections)

Quantify

- bandwidth or throughput: total work done in a given time
- latency or response time: time between start and completion of an event
- example: in a pipelined processor, it’s running a series of add instruction. There might be multiple add running at the same time.
    - throughput: how many add can be issued in a given time
    - latency: for one add instruction, how long does it take from issue to complete.

- bandwidth outpaced latency

Feature size

- transistor size in x or y dimension
- transistor density increases
- wire latency do not increase that fast

Trends in Power and Energy in ICs

Power and Energy

power is unit time energy

Thermal Design Power (TDP)

- characterize sustained power consumption
- use as target for power supply and cooling
- lower than peak power, higher than average power consumption

Clock Rate

can be reduced dynamically to reduce power consumption

Dynamic Energy and Power

- dynamic energy per transistor
  - used for a transistor switching from 0 to 1 / 1 to 0
  - 1/2capacitive load × voltage2
  - related to number of transistor and technology
- dynamic power per transistor
  - 1/2capacitive load × voltage2 × frequency switched
- voltage is the key
  - voltage of processor has become lower

Power

- 130W, maximum for air cooling
- we use energy of a specific task to compare CPUs

Techniques

- turn off clock
- dynamic voltage-frequency scaling
- low power state for DRAM, disks
- overclocking, turning off cores

Static Power

- current × voltage
- scales with number of transistors
- power gating: turning off the power supply

Trends in Cost

- refer to slides and textbook
- I don’t want to type exactly the same thing in slides
- From my perspective, quantifying these things is the core of the book CAAQA. However, this is not our course orients to.

Dependability

- MTTF ~ Exp(n)
- MTTF Components ~ Exp(n1 + n2 +...)
- $E(Exp(x)) = \frac{1}{x}$
- Serial MTTF

$$\frac{1}{\frac{1}{MTTF} + \frac{1}{MTTF}}$$

- Redundant MTTF

$$\frac{MTTF^2}{2 \times MTTR}$$

Measuring Performance

Typical Performance Metrics

- latency (response time)
- throughput (bandwidth)

Benchmarks

- a common program for testing the execution times of computers
- e.g. kernels (matrix multiply), top programs (e.g. sorting), synthetic benchmarks
- we use benchmark suite (SPEC)
- SPEC uses performance ratio
- and Geometric Mean

Quantitative Principles

Principles for Computer Design

> take advantage of parallelism
> principle of locality
> focus on common case

Amdahl’s Law

- performance improvement of using a new feature is limited by the fraction of the time the new feature can be used
- formula

$$\text{Execution time}_{\text{new}} = \text{Execution time}_{\text{old}} \times \left((1 - \text{Fraction}_{\text{enhanced}}) + \frac{\text{Fraction}_{\text{enhanced}}}{\text{Speedup}_{\text{enhanced}}} \right)$$

$$\text{Speedup}_{\text{overall}} = \frac{\text{Execution time}_{\text{old}}}{\text{Execution time}_{\text{new}}} = \frac{1}{(1 - \text{Fraction}_{\text{enhanced}}) + \frac{\text{Fraction}_{\text{enhanced}}}{\text{Speedup}_{\text{enhanced}}}}$$

  - specific feature can speedup the program. **fraction is the part that actually accelerated (10%...). Speedup is the times that the acceleration (2 times ...)**
  - can compute fraction, speedup.